

**ATMIYA UNIVERSITY**

**RAJKOT**



A

Report On

**AI-Powered Resume Screening System Dashboard**

Under subject of

**MINI PROJECT**

B. TECH, Semester – VII

**(Computer Engineering)**

Submitted by:

Riddhi Sanjaybhai Khokhariya (Enrollment No: 220002040)

**Prof. Devangi Rasikbhai Paneri**

(Faculty Guide)

**Prof. Tosal M. Bhalodia**

(Head of the Department)

Academic Year

**(2025-26)**

## **CANDIDATE'S DECLARATION**

We hereby declare that the work presented in this project entitled “**AI-Powered Resume Screening System Dashboard**” submitted towards completion of project in **7<sup>th</sup> Semester** of B. Tech. (Computer Engineering) is an authentic record of our original work carried out under the guidance of “**Devangi Rasikbhai Paneri**”.

We have not submitted the matter embodied in this project for the award of any other degree.

Semester: 7<sup>th</sup>

Place: Rajkot

**Signature:**

Riddhi Khokhariya (220002040)

---

**ATMIYA UNIVERSITY**  
**RAJKOT**



**CERTIFICATE**

**Date:** 13-09-2025

This is to certify that the “**AI-Powered Resume Screening System Dashboard**” has been carried out by **Riddhi Khokhariya** under my guidance in fulfillment of the subject Mini Project in COMPUTER ENGINEERING (7<sup>th</sup> Semester) of Atmiya University, Rajkot during the academic year 2025.

Prof. Devangi Rasikbhai Paneri

Prof. Tosal M. Bhalodia

**(Project Guide)**

**(Head of the Department)**



## **ACKNOWLEDGEMENT**

We have taken many efforts in this project. However, it would not have been possible without the kind support and help of many individuals and organizations. We would like to extend our sincere thanks to all of them.

We are highly indebted to **Prof. Devangi Rasikbhai Paneri** for their guidance and constant supervision as well as for providing necessary information regarding the Project titled “**AI-Powered Resume Screening System Dashboard**”. We would like to express our gratitude towards staff members of the Computer Engineering Department, Atmiya University for their kind co-operation and encouragement which helped us in completion of this project.

We even thank and appreciate our colleague in developing the project and people who have willingly helped us out with their abilities.

Riddhi Khokhariya (220002040)

---

## **ABSTRACT**

This project presents an AI-powered Resume Screening System Dashboard designed to automate and simplify the candidate shortlisting process using Natural Language Processing (NLP). The system extracts key details such as name, email, phone number, and skills from resumes and compares them against a required skill set defined by recruiters. Candidates are ranked based on their skill match, and results are displayed on an interactive dashboard with visual charts and downloadable reports.

The system includes provisions for dynamic skill set management, allowing recruiters to update job-specific requirements easily. A database is integrated to securely store uploaded resumes, enabling future retrieval and analysis. This solution enhances recruitment efficiency by reducing manual effort, minimizing human error, and ensuring faster and more reliable shortlisting decisions.

---

## INDEX

Sr. No.	TITLES		Page No.
	<b>Acknowledgement</b>		-
	<b>Abstract</b>		-
	<b>List of Figures</b>		<b>1</b>
	<b>List of Tables</b>		<b>2</b>
<b>1.</b>	<b>Introduction</b>		<b>3</b>
	1.1	Purpose	3
	1.2	Scope	3
	1.3	Technology and tool	3
<b>2.</b>	<b>Project Management</b>		<b>4</b>
	2.1	Project Planning	4
	2.2	Project Scheduling	4
	2.3	Risk Management	5
	2.3.1	Risk Identification	5
	2.3.2	Risk Analysis	5
<b>3.</b>	<b>System Requirements Study</b>		<b>6</b>
	3.1	Hardware and Software Requirements	6
	3.1.1	Server side hardware requirement	6
	3.1.2	Software requirement	6
	3.1.3	Client Side requirement	6
	3.2	Constraints	7
	3.2.1	Hardware Limitation	7
	3.2.2	Reliability requirements	7
	3.2.3	Safety and Security Consideration	7
<b>4.</b>	<b>System Analysis</b>		<b>8</b>
	4.1	Study of Current System	8
	4.2	Problem and Weaknesses of Current System	8
	4.3	Requirements of New System	8
	4.3.1	User Requirements	8
	4.3.2	System Requirements	9

	4.4	Feasibility Study	9
	4.5	Feature Of New System	9
<b>5</b>	<b>System Design</b>		<b>10</b>
	5.1	Input /output interface	10
	5.2	Interface Design	10
	5.2.1	Class Diagram	11
	5.2.2	Use Case Diagram	11
	5.2.3	Activity Diagram	12
	5.2.4	Data Flow Diagram	12
	5.2.5	State Diagram	13
	5.2.6	E-R Diagram	13
	5.2.7	Sequence Diagram	13
<b>6</b>	<b>Code Implementation</b>		<b>14</b>
	6.1	Implementation Environment	14
	6.2	Program/Module Specification	14
	6.3	Coding Standards	14
<b>7</b>	<b>Testing</b>		<b>15</b>
	7.1	Testing Strategy	15
	7.2	Testing Method	15
	7.2.1	Unit Testing	15
	7.2.2	Integration Testing	15
	7.2.3	Validation Testing	15
	7.3	Test Cases	15
	7.3.1	Test Suite	15
<b>8</b>	<b>Limitations and Future Enhancement</b>		<b>16</b>
	8.1	Limitations	16
	8.2	Future Enhancement	16
<b>9</b>	<b>Conclusion</b>		<b>17</b>
<b>10</b>	<b>References</b>		<b>18</b>

## **LIST OF FIGURES**

<b>Fig. No.</b>	<b>Figure Title</b>	<b>Page No.</b>
<b>1.1</b>	Class Diagram	11
<b>1.2</b>	Use Case Diagram	11
<b>1.3</b>	Activity Diagram	12
<b>1.4</b>	Data Flow Diagram	12
<b>1.5</b>	State Diagram	13
<b>1.6</b>	ER Diagram	13
<b>1.7</b>	Sequence Diagram	13
<b>1.8</b>	Dashboard Screenshot	10
<b>1.9</b>	Candidate Ranking Chart	10



## **LIST OF TABLES**

<b>Table No.</b>	<b>Table Title</b>	<b>Page No.</b>
<b>3.1.1</b>	Server-side Hardware Requirements	6
<b>3.1.2</b>	Software Requirements	6
<b>3.1.3</b>	Client-side Requirements	6

# **CHAPTER – 1**

## **INTRODUCTION**

### **1.1. Purpose**

The purpose of this project is to automate resume screening using AI and NLP techniques, reducing recruiter workload and increasing accuracy in candidate shortlisting.

### **1.2 Scope**

- Resume parsing using NLP.
- Skill extraction and candidate ranking.
- Interactive dashboard for visualization.
- Dynamic skill set management by recruiters.
- Secure resume storage in a database.

### **1.3 Technology and Tools**

- **Frontend:** HTML5, CSS3, Bootstrap 5, JavaScript
- **Backend:** Python (Flask/Django)
- **Libraries:** NLP (NLTK, spaCy), Pandas, Chart.js, DataTables
- **Database:** MySQL/PostgreSQL
- **IDE/Editor:** VS Code, Jupyter Notebook

## **CHAPTER – 2**

### **PROJECT MANAGEMENT**

#### **2.1 Project Planning**

The project was carried out in a structured manner and divided into five main phases: requirement analysis, system design, implementation, testing, and deployment. In the requirement analysis phase, the needs of recruiters were studied, and the essential features of the system were identified. The design phase focused on preparing the architecture and diagrams to provide a clear workflow. The implementation phase involved developing modules for resume parsing, ranking, and dashboard integration. Testing was then conducted to ensure accuracy and system performance, followed by the final deployment for real-world usage. This phased approach ensured smooth execution and minimized complexity.

#### **2.2 Project Scheduling**

The project was completed over 10 weeks:

- **Week 1–2:** Requirement study and research.
- **Week 3–4:** System design and database schema.
- **Week 5–7:** Backend implementation (parsing, ranking, storage).
- **Week 8–9:** Dashboard development and integration.
- **Week 10:** Testing and deployment.

## 2.3 Risk Management

Risk management is an essential part of the project to identify potential challenges and take preventive measures. It helps ensure that the system remains accurate, secure, and efficient during real-world use.

### 2.3.1 Risk Identification:

1. **Errors in skill extraction** – Resumes often contain ambiguous text, synonyms, or abbreviations (e.g., “ML” vs. “Machine Learning”), which may lead to inaccurate skill extraction.
2. **Privacy concerns while storing resumes** – Candidate resumes contain sensitive data such as personal details, academic background, and work history, which could be misused if not stored securely.
3. **Scalability issues with large datasets** – When a large number of resumes are uploaded, the system may face delays, performance degradation, or storage problems.

-

### 2.3.2 Risk Analysis:

4. **Improving accuracy with NLP models** – Advanced Natural Language Processing (NLP) techniques such as lemmatization, synonym mapping, and named entity recognition help reduce extraction errors.
5. **Ensuring data privacy** – A secure database with encryption and role-based access control is implemented to protect candidate information from unauthorized access.
6. **Enhancing scalability** – Database indexing, optimized queries, and efficient storage mechanisms ensure smooth processing even with bulk resume uploads.

## **CHAPTER – 3**

### **SYSTEM REQUIREMENTS STUDY**

#### **3.1 Hardware and Software Requirement**

##### **3.1.1 Hardware Requirements:**

###### **Server-side Hardware Requirement:**

<b>Devices</b>	<b>Description</b>
Processor	Intel i5/i7 or Higher
RAM	8 GB Minimum
Storage	100 GB SSD
Operating System	Windows / Linux

Table 3.1.1 Server-side Hardware Requirement

##### **3.1.2 Software Requirements:**

<b>For Which</b>	<b>Software*</b>
Programming Language	Python 3.x
Framework	Flask / Django Framework
Database	MySQL / PostgreSQL
Front End	Bootstrap 5, Chart.js

Table 3.1.2 Software Requirement

##### **3.1.3 Client-side Requirements:**

<b>For Which</b>	<b>Requirement*</b>
Web Browser	Modern Browser (Chrome, Edge, Firefox)
Internet	Stable Internet Connectivity

Table 3.1.3 Client-side Requirement

## **3.2 Constraints**

### **3.2.1 Hardware Limitations:**

The performance of the system is dependent on the available server resources. Limited server capacity, such as lower processing power, insufficient RAM, or restricted storage, may affect response times during bulk resume uploads. This could lead to delays in parsing and ranking operations when handling large datasets.

### **3.2.2 Reliability Requirements:**

The system must ensure consistent performance even under high workloads. It should be able to handle multiple resumes being uploaded and processed concurrently without system crashes or data loss. Reliability is essential to maintain user trust and ensure smooth functioning during peak usage.

### **3.2.3 Safety and Security Consideration:**

Since resumes contain sensitive candidate information, data security is a primary concern. All resumes must be stored in a secure database with encryption, and access should be restricted only to authorized users. Proper security measures, such as authentication and secure connections, should be enforced to prevent unauthorized access or data breaches.

## **CHAPTER – 4**

### **SYSTEM ANALYSIS**

#### **4.1 Study Current System**

In most organizations, resume screening is performed manually by recruiters or HR staff. This process involves going through each resume individually, identifying key skills, and shortlisting candidates for further evaluation. While this approach works for small-scale recruitment, it becomes inefficient and time-consuming when handling hundreds or thousands of applications. Manual screening is also prone to human error and subjectivity, which may lead to overlooking suitable candidates.

#### **4.2 Problem and weakness of current system**

- High dependency on manual effort
- Time-consuming and error-prone process
- Inconsistency in evaluation due to subjective judgment
- Risk of missing out on qualified candidates
- Difficulty in handling large volumes of resumes

#### **4.3 Requirements of New System**

##### **4.3.1 User Requirements:**

- Automated extraction of candidate details and skills
- Ability to upload resumes in common formats (PDF/DOCX)
- Easy-to-use dashboard for recruiters to view results
- Downloadable reports for further analysis

### 4.3.2 System Requirements:

- Resume parsing module using NLP techniques
- Candidate ranking mechanism based on job-specific skills
- Database for secure storage and retrieval of resumes
- Visualization tools for displaying rankings and insights

### 4.4 Feasibility Study

- **Technical Feasibility:** The system is technically feasible as NLP libraries (NLTK, spaCy) and web frameworks (Flask/Django) are readily available. Existing hardware and software resources are sufficient to run the system.
- **Operational Feasibility:** The system improves recruitment efficiency, reduces workload, and ensures consistency in candidate evaluation. Recruiters can easily adapt to its user-friendly dashboard.
- **Economic Feasibility:** By reducing the time and cost involved in manual screening, the system provides a cost-effective solution. Maintenance costs are minimal compared to traditional recruitment expenses.

### 4.5 Features of New System

- Automated resume parsing using AI and NLP
- Dynamic skill set definition for recruiters
- Candidate ranking based on skill match
- Interactive dashboard with charts and reports
- Secure storage of resumes in an encrypted database

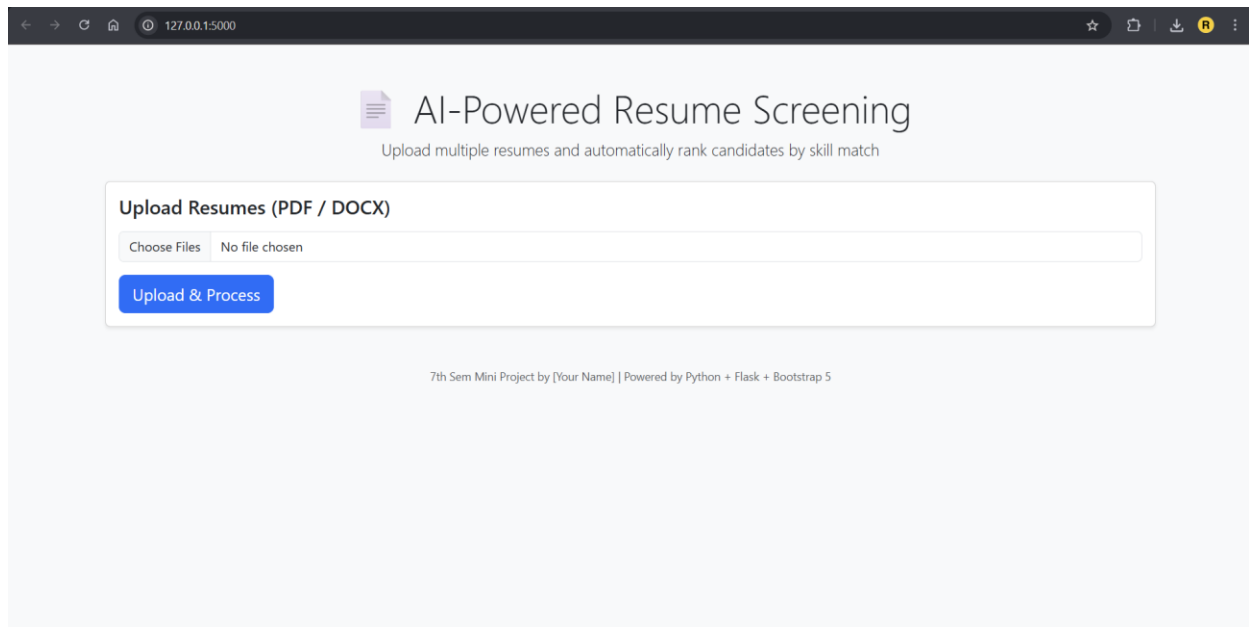


# CHAPTER – 5

## System Design

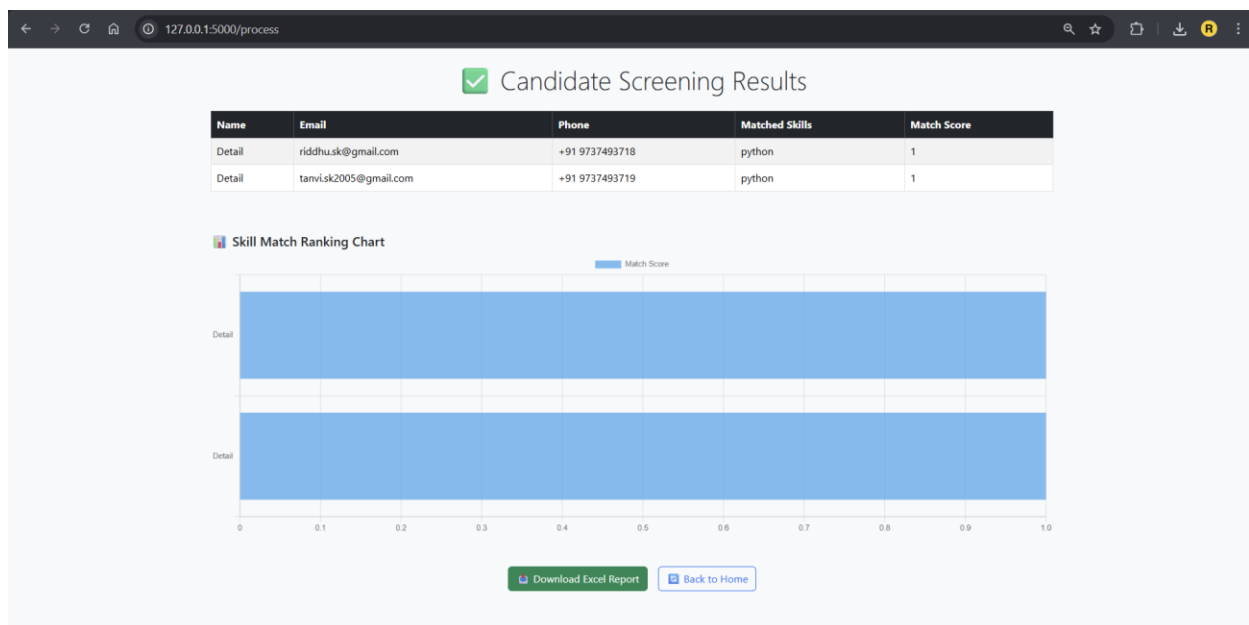
### 5.1 Input/Output Interface

- **Input:** Resumes uploaded in PDF or DOCX format, job skill requirements provided by recruiters



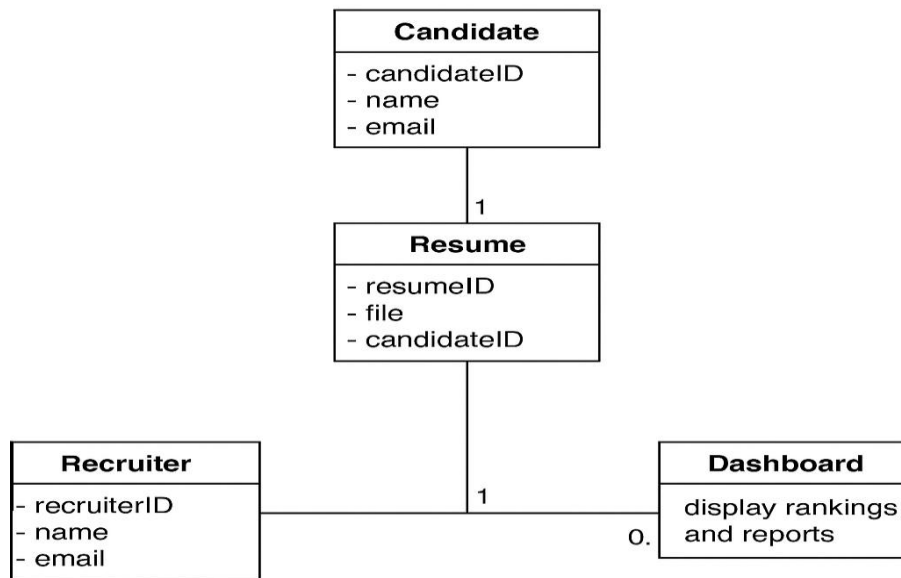
The screenshot shows a web browser window with the address bar displaying '127.0.0.1:5000'. The page title is 'AI-Powered Resume Screening' with a subtitle 'Upload multiple resumes and automatically rank candidates by skill match'. Below the title is a file upload section labeled 'Upload Resumes (PDF / DOCX)' containing a 'Choose Files' button, a text box showing 'No file chosen', and an 'Upload & Process' button. At the bottom of the page, a footer reads '7th Sem Mini Project by [Your Name] | Powered by Python + Flask + Bootstrap 5'.

- **Output:** Candidate ranking list, skill match reports, charts on dashboard

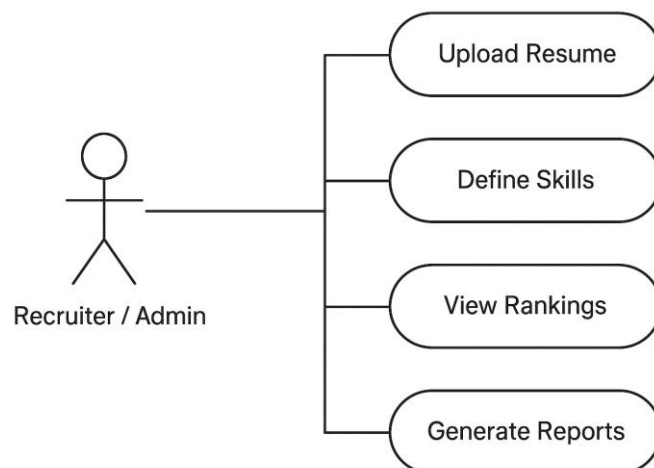


## 5.2 Interface Design

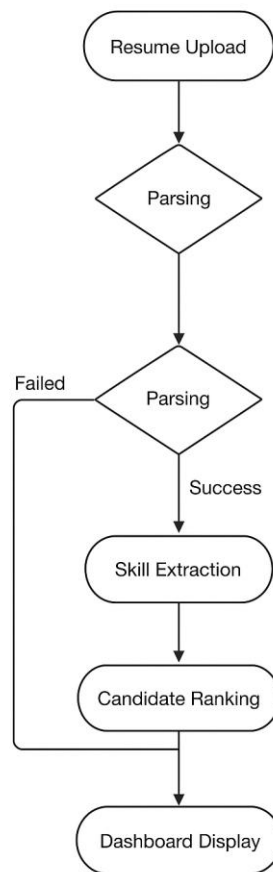
### 5.2.1 Class Diagram:



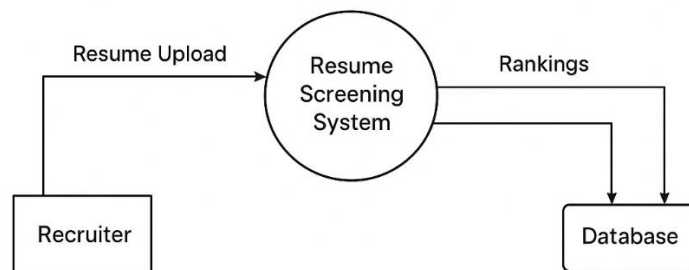
### 5.2.2 Use Case Diagram:



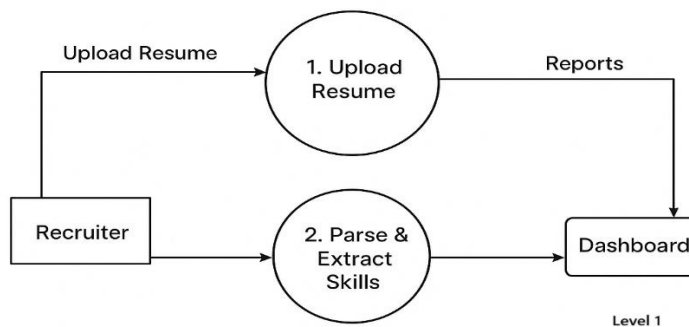
### 5.2.3 Activity Diagram:



### 5.2.4 Data Flow Diagram:

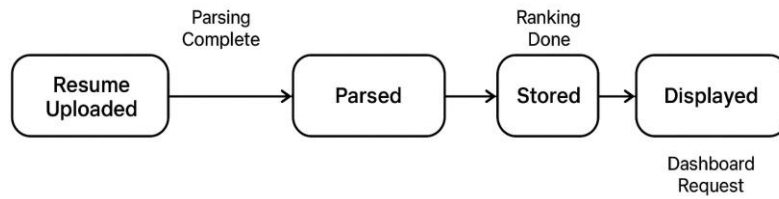


Level-0

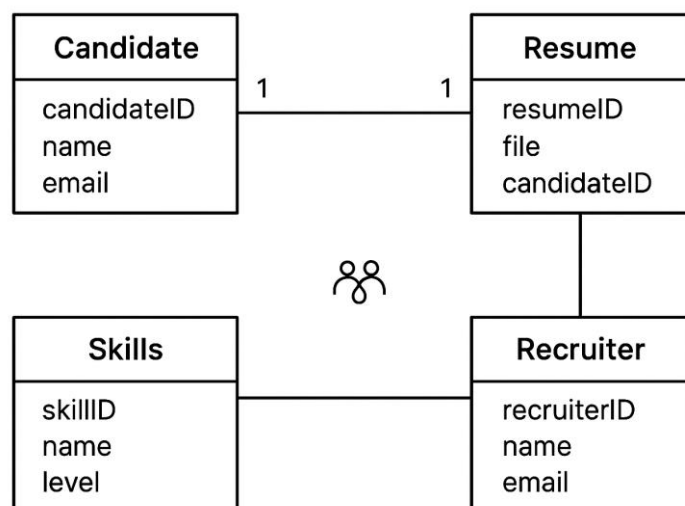


Level 1

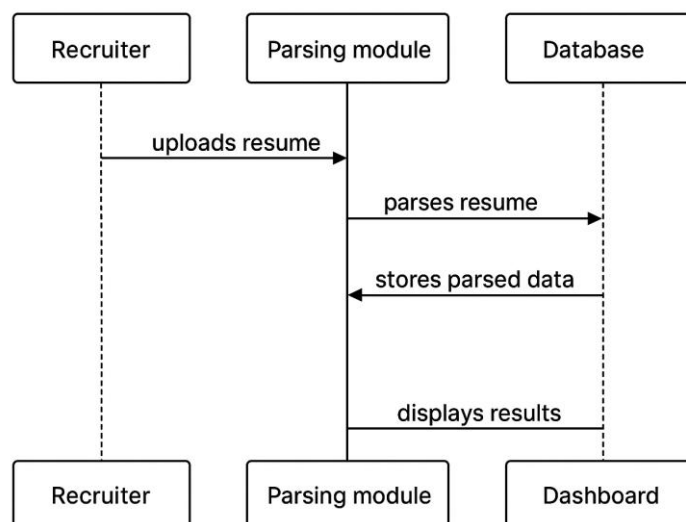
### 5.2.5 State Diagram:



### 5.2.6 E-R Diagram:



### 5.2.7 Sequence Diagram:



# **CHAPTER – 6**

## **Code Implementation**

### **6.1 Implementation Environment**

The project was developed using Python with Flask as the backend framework. NLP libraries such as NLTK and spaCy were used for text extraction and processing. The frontend was built using Bootstrap 5, Chart.js, and DataTables for visualization. MySQL was used as the database for storing parsed resume data.

### **6.2 Program/Module Specification**

- **Resume Parser Module** – Extracts text and candidate details from resumes.
- **Skill Extraction Module** – Identifies and maps skills against the required skill set.
- **Ranking Module** – Calculates candidate scores and ranks them.
- **Dashboard Module** – Displays results with interactive charts and tables.
- **Database Module** – Handles secure storage and retrieval of resumes.

### **6.3 Coding Standards**

- Code written following **PEP 8 guidelines** for Python.
- Modular coding approach for maintainability.
- Use of exception handling to avoid runtime errors.
- Inline comments and documentation for clarity.

## **CHAPTER – 7**

### **Testing**

#### **7.1 Testing Strategy**

The testing strategy combined unit testing, integration testing, and validation testing to ensure the reliability of the system. Each module was first tested individually, then integrated, and finally validated with real sample resumes.

#### **7.2 Testing Method**

##### **7.2.1 Unit Testing:**

Each module, such as resume parsing, skill extraction, and ranking, was tested individually to ensure correct functionality before integration.

##### **7.2.2 Integration Testing:**

The interaction between modules was tested by checking the flow from resume upload to parsing, storing in the database, ranking, and displaying results on the dashboard.

##### **7.2.3 Validation Testing:**

The integration testing is meant to test all the modules simultaneously because it is possible that all the modules may function correctly when tested individually. But they may not work altogether and may lead to unexpected outcome.

#### **7.3 Test Cases**

##### **7.3.1 Test Suite**

- Uploading valid and invalid resume formats
- Testing skill extraction accuracy
- Verifying ranking output consistency
- Checking dashboard visualization and report generation

## **CHAPTER – 8**

### **Limitations and Future Enhancement**

#### **8.1 Limitations**

- The system shows limited accuracy when dealing with resumes that are poorly formatted or scanned as images.
- It relies on the skill sets defined by recruiters, which means results may vary if the required skills are not comprehensive.
- Currently, the system supports only English resumes, restricting its use in multilingual recruitment scenarios.

#### **8.2 Future Enhancement**

- Adding multi-language support will expand the system for global use.
- Integration with job portals can automate resume collection and reduce manual effort.
- AI-based recommendations can suggest additional skills to improve candidate ranking.
- Enhanced analytics and visualization will make the dashboard more informative for recruiters.

## **CHAPTER – 9**

### **Conclusion**

The AI-Powered Resume Screening System Dashboard offers an effective and reliable solution to overcome the challenges of traditional recruitment methods. By automating key processes such as resume parsing, skill extraction, and candidate ranking, it significantly reduces the manual effort required from recruiters while improving accuracy and consistency in shortlisting. The system's dynamic skill set management allows recruiters to tailor the screening process according to specific job roles, making it more adaptable to varying requirements.

In addition, the interactive dashboard with charts and reports provides a clear visual representation of candidate suitability, helping recruiters make informed, data-driven decisions. While the system currently has limitations, such as dependence on recruiter-defined skill sets and support for only English resumes, it lays a strong foundation for future growth. With enhancements like multi-language support, job portal integration, and advanced analytics, the system can evolve into a comprehensive recruitment tool capable of meeting global hiring demands.



## **CHAPTER – 10**

### **Reference**

1. Bird, Steven, Edward Loper, and Ewan Klein. *Natural Language Processing with Python*. O'Reilly Media.
2. Miguel Grinberg. *Flask Web Development*. O'Reilly Media.
3. Silberschatz, Abraham, Henry Korth. *Database System Concepts*. McGraw-Hill.
4. Bootstrap Documentation – <https://getbootstrap.com>
5. Chart.js Documentation – <https://www.chartjs.org>
6. W3Schools – <https://www.w3schools.com>